

## How to use the ID Issuer API

If you want to use the API of the ID issuer test instance, you have to authenticate yourself using OpenID Connect (OIDC). You will also need a TLS client certificate which will be sent to you in an e-mail following registration. The pertinent password will be delivered via SMS.

The following URL must be called to get the OIDC access token:

Public test system:

<https://keycloak.test.de.id-issuer.eu/auth/realms/idissuer/protocol/openid-connect/token>

Productive system:

<https://iam.de.id-issuer.eu/auth/realms/idissuer/protocol/openid-connect/token>

The POST request for the access token must contain the parameters stated below:

Parameter	Value
clientId	Fixed value "self-management-gui"
clientUsername	Can be obtained from the self-management portal
clientPassword	Can be obtained from the self-management portal
clientSecret	Can be obtained from the self-management portal
grantType	Fixed value "password"

### Example of getting an access token with CURL:

Initial:

```
curl -d "grant_type=password&username=[obtained from the SP]&password=[obtained from the SP]&client_id=self-management-gui&client_secret=[obtained from the SP]" -H "Content-Type: application/x-www-form-urlencoded" -X POST "https://keycloak.test.de.id-issuer.eu/auth/realms/idissuer/protocol/openid-connect/token"
```

Filled with values:

```
curl -d "grant_type=password&username=Uhhkzzh6&password=Jgfou75!&client_id=self-management-gui&client_secret=3d308e29-13d6-46eb-be3b-567ac5275277" -H "Content-Type: application/x-www-form-urlencoded" -X POST "https://keycloak.test.de.id-issuer.eu/auth/realms/idissuer/protocol/openid-connect/token"
```

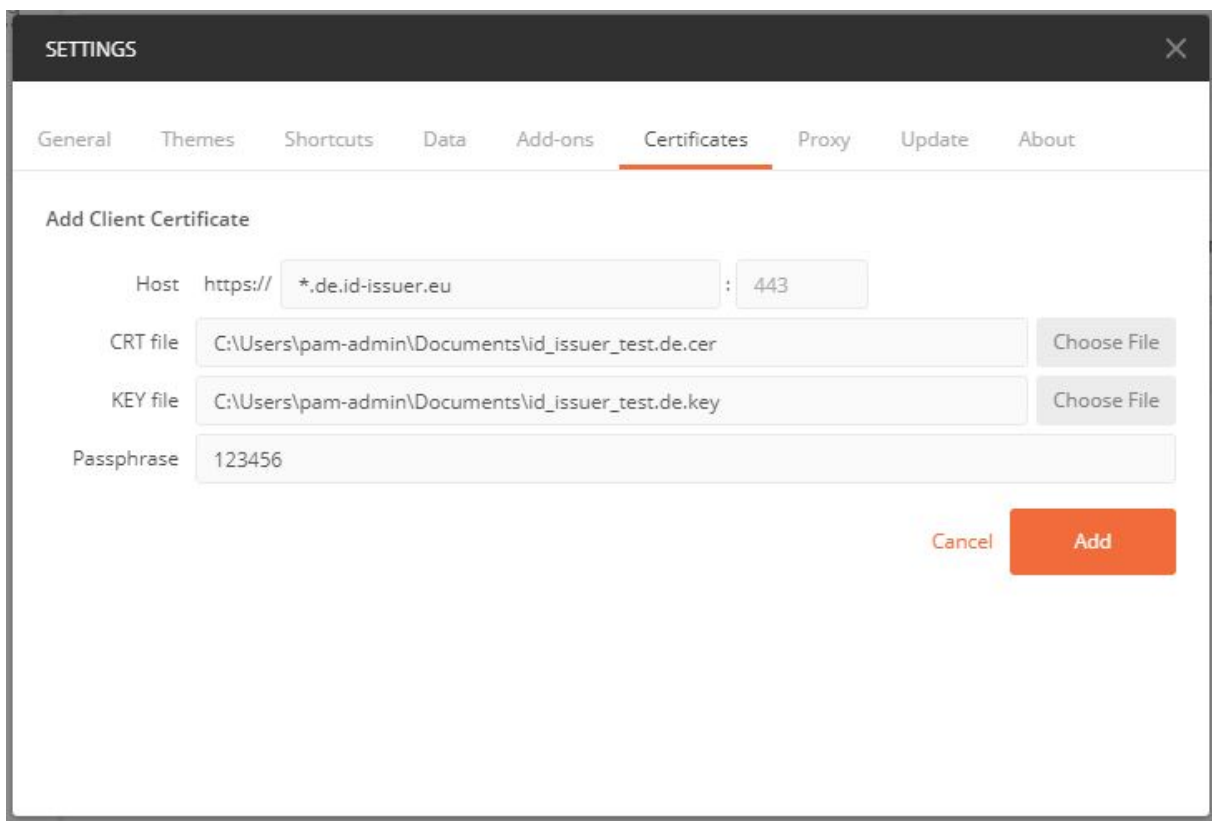
## Using the API with Postman

Two things are required when using the API with Postman:

On the one hand, the TLS client certificate must be added and on the other hand, the access token must be obtained. It is also possible to directly import the YAML file as a collection into Postman.

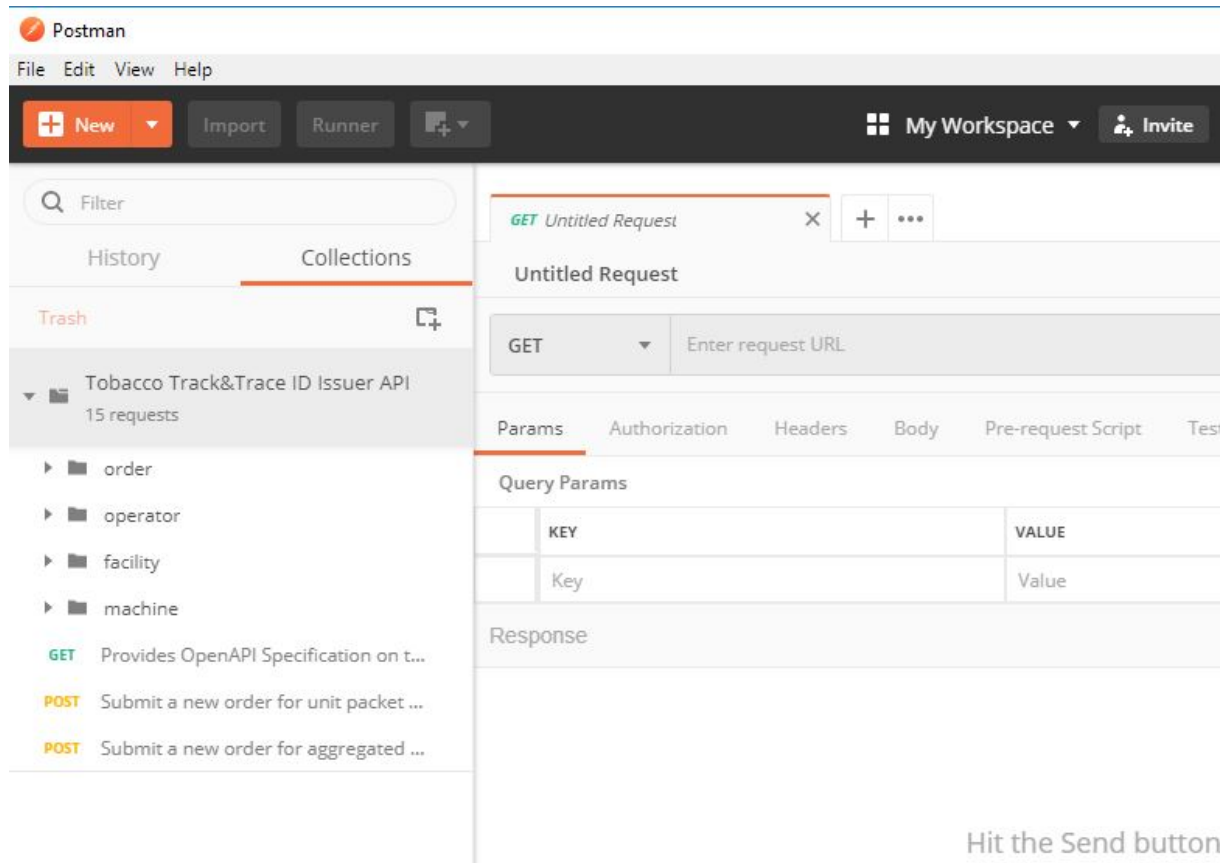
### 1. Adding the TLS client certificate in Postman

The Postman tool cannot use P12 containers. Therefore, the private key must be exported from the P12 container as PKCS8 and the certificate chain in PEM format (base64). You can then add the TLS client certificate in the Postman settings as shown below.



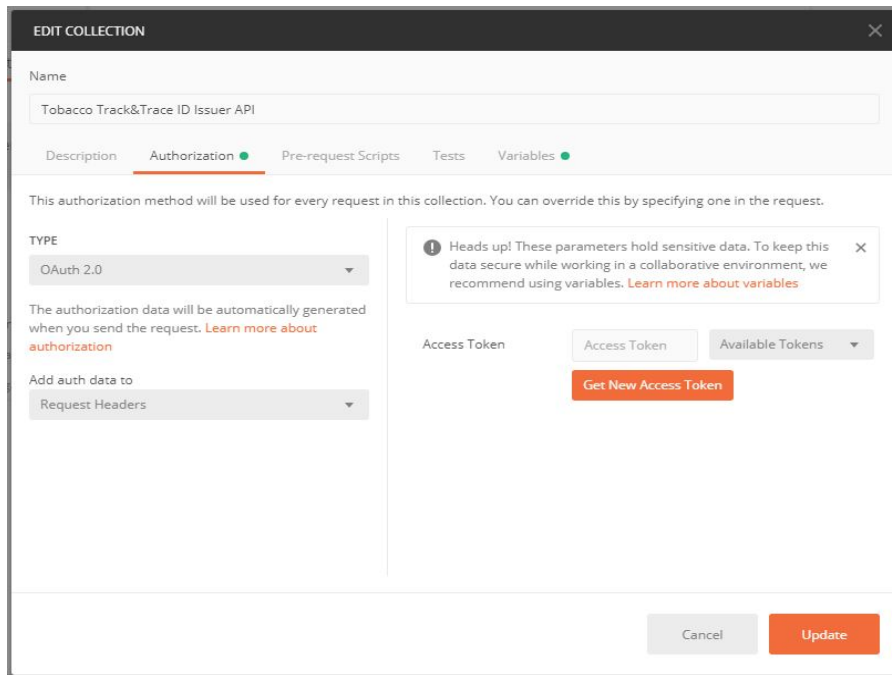
## 2. Importing the YAML file

The YAML file can be imported and you will then have a collection that shows all the requests for the ID issuer instance.

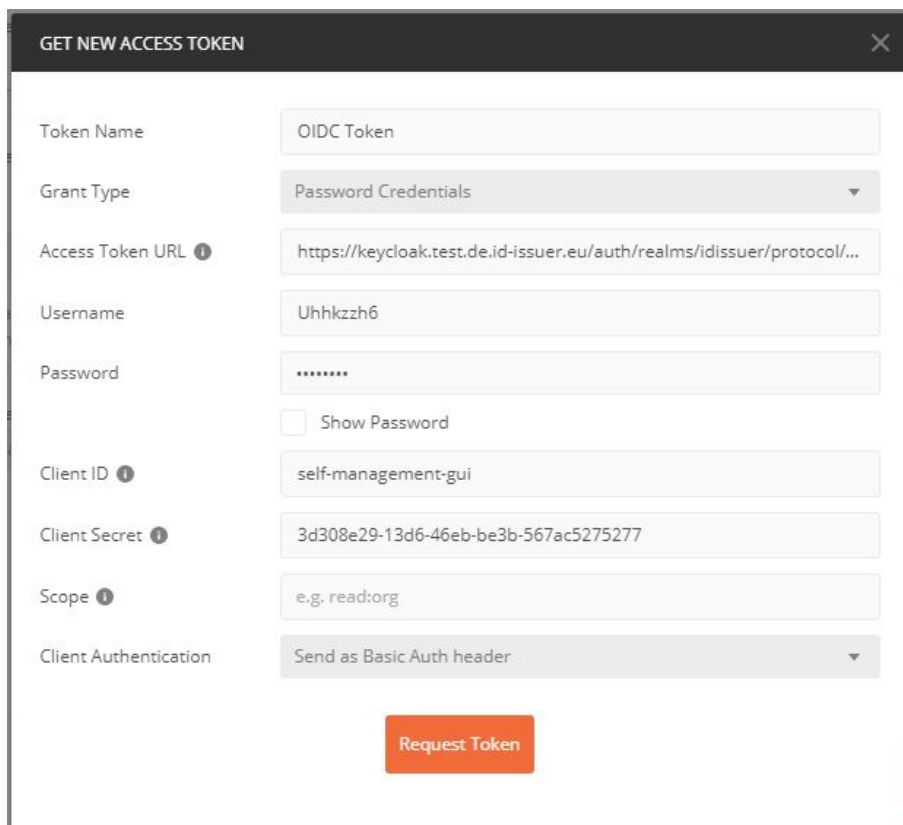


## 3. Providing the access token for the collection

In order to be able to use the requests, the authorization for the collection must first be set. Click the collection with the right mouse button and select "Edit" in the context menu. Then click the Authorization tab and select the "OAuth2.0" authorization type.

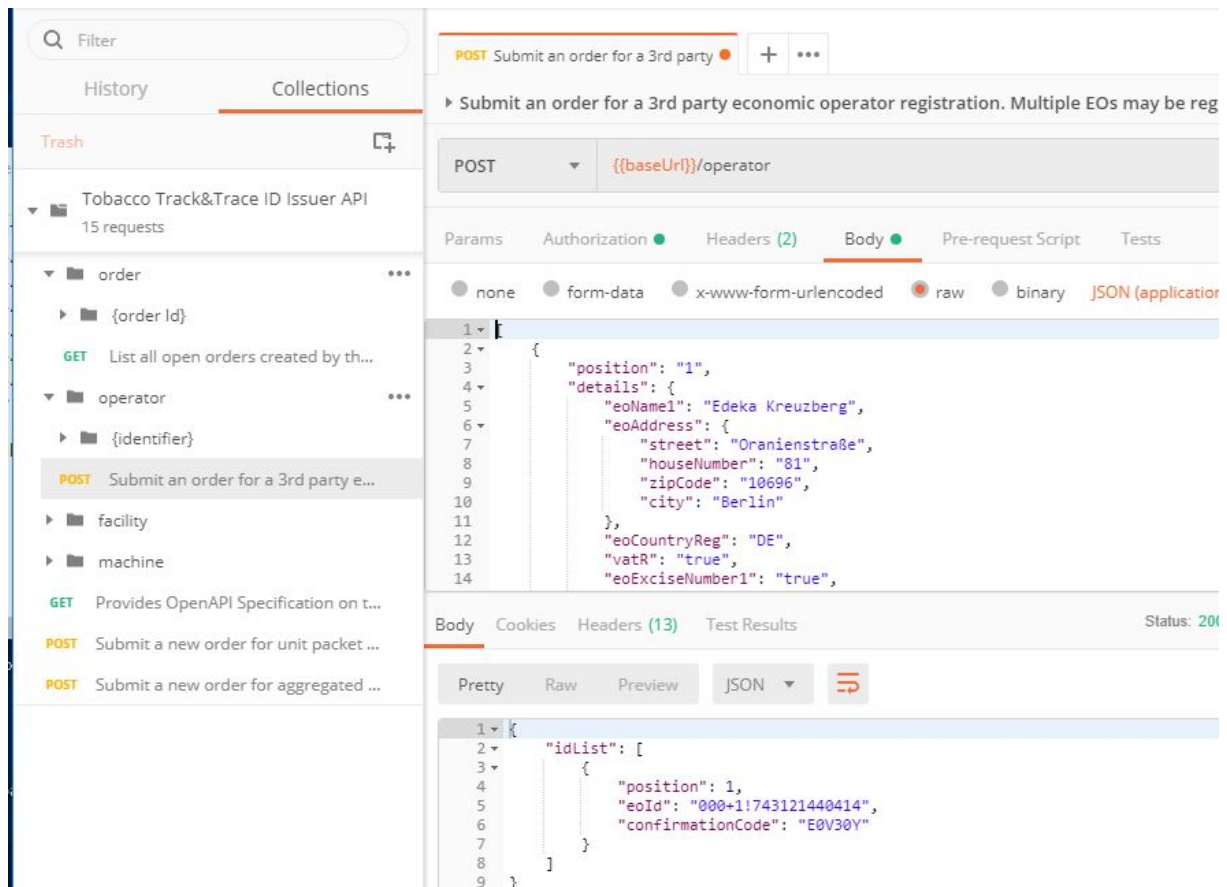


Now click the "Get New Access Token" button and enter the parameters for the access token. The parameters were described earlier in this document. An example is given in the following illustration.



#### 4. Example: performing 3rd party registration

In order to make a request, call the request from the collection, for instance, 3rd party registration. To execute the request, the access token called must be set in the Authorization tab and the request must be filled with values in the Body tab.



The screenshot displays the Postman interface for a REST client. On the left, a sidebar shows a collection named "Tobacco Track&Trace ID Issuer API" with 15 requests. The "operator" collection is expanded, showing a "POST" request titled "Submit an order for a 3rd party e...".

The main interface shows the details of this request:

- Method:** POST
- URL:** `{{baseUrl}}/operator`
- Body:** raw (application/json)
- Request Body (JSON):**

```
1 {
2   "position": "1",
3   "details": {
4     "eoName1": "Edeka Kreuzberg",
5     "eoAddress": {
6       "street": "Oranienstraße",
7       "houseNumber": "81",
8       "zipCode": "10696",
9       "city": "Berlin"
10    },
11    "eoCountryReg": "DE",
12    "vatR": "true",
13    "eoExciseNumber1": "true",
14  }
}
```

The response is shown in the "Body" tab, rendered as JSON:

```
1 {
2   "idList": [
3     {
4       "position": 1,
5       "eoId": "000+1!743121440414",
6       "confirmationCode": "E0V30V"
7     }
8   ]
9 }
```

The status bar at the bottom right indicates "Status: 200".